TEAM H: Progress Meeting

Donghun Jung

Chanyoung Lee

Yujin Seo

13.Oct.2023

Sungkyunkwan University

Remind Our Project

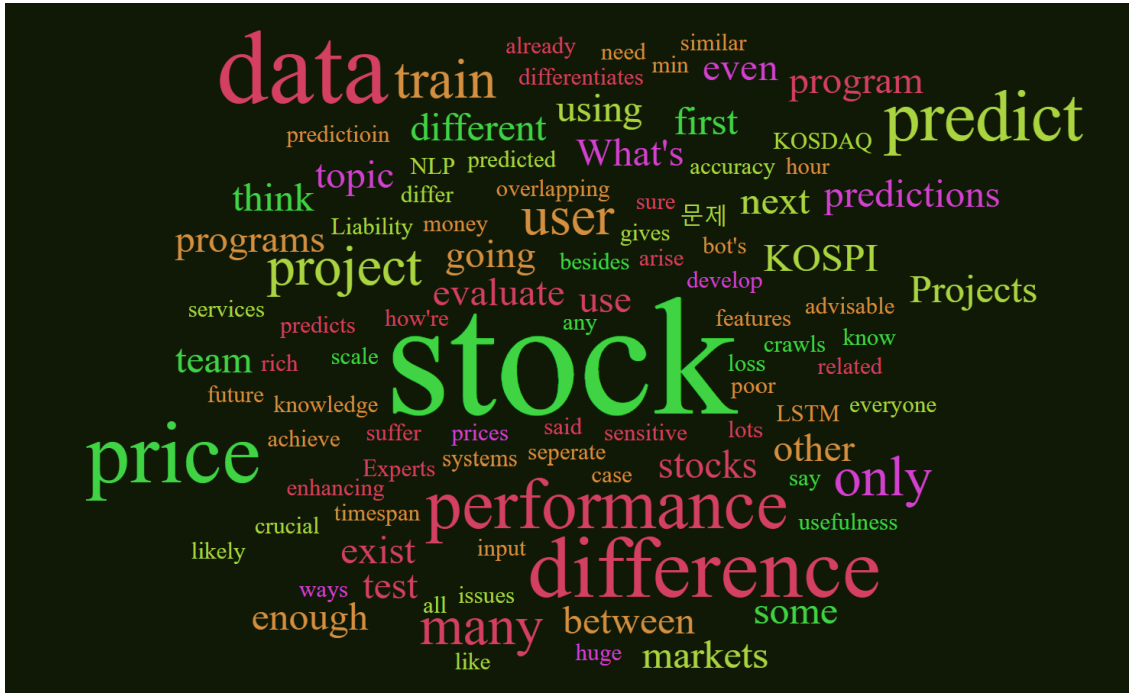
Stock Price Prediction Systems

- Not real time prediction
- Predict price of SAMSUNG Electronics

Feedbacks

What's the difference between our project and other services?

Predicting stock prices is difficult.



What's the difference?

Predicting stock prices is difficult.

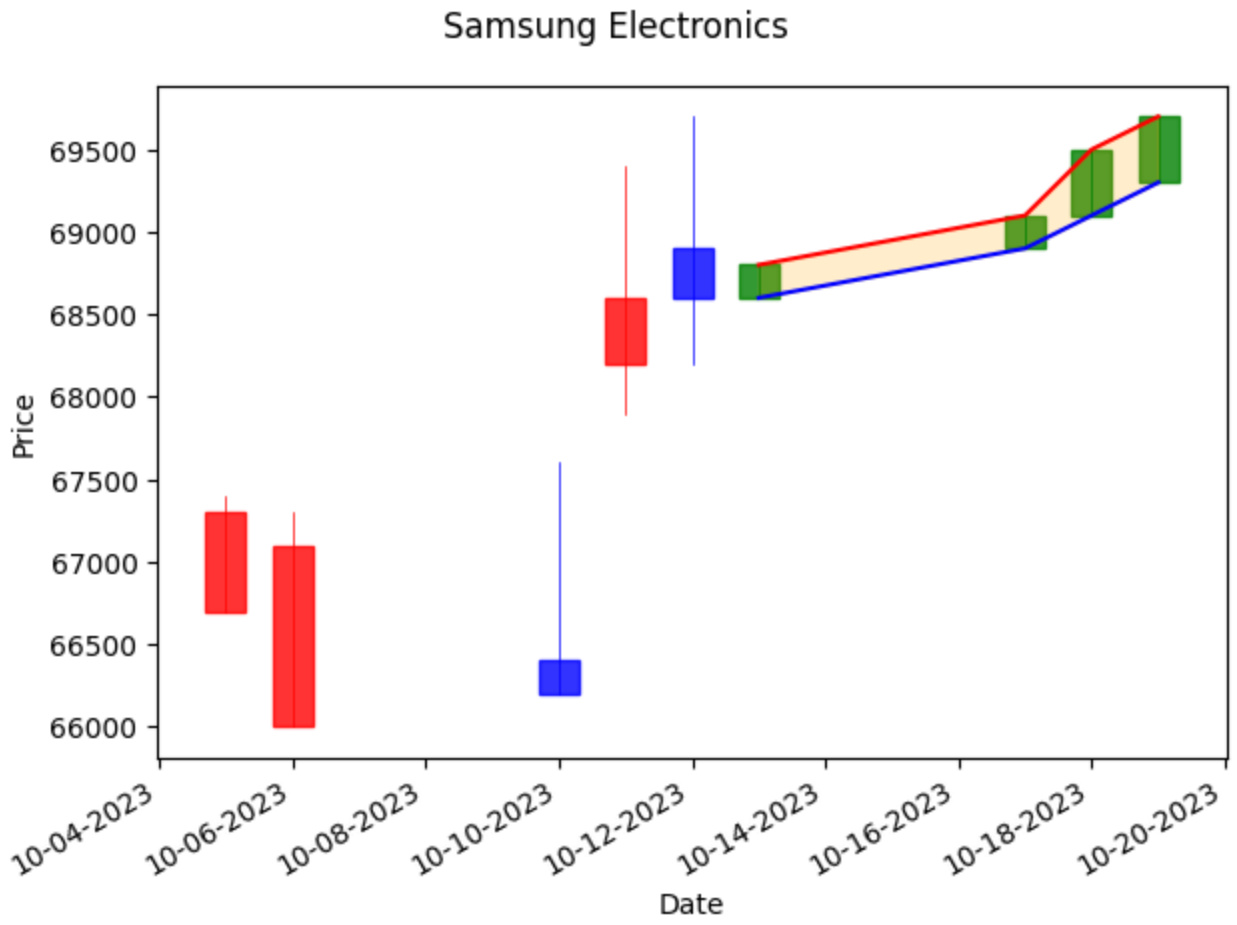
손익계산서	23.06/30	23.03/31	22.12/31	22.09/30	22.06/30
매출액	94,028	86,035	81,462	74,863	67,166
영업이익	12,652	12,717	13,656	12,368	10,684
EBITDA	17,648	16,870	17,626	16,173	14,295
영업외이익	847	335	254	232	235
순이익	12,195	11,751	12,556	11,190	9,516

재무상태표	23.06/30	23.03/31	22.12/31	22.09/30	22.06/30
자산총계	90,591	86,833	82,338	74,426	68,513
부채총계	38,409	37,598	36,440	33,302	30,855
자본총계	51,130	48,054	44,704	39,851	36,376
주식수(만주)	317,100	316,600	313,000	314,600	311,100

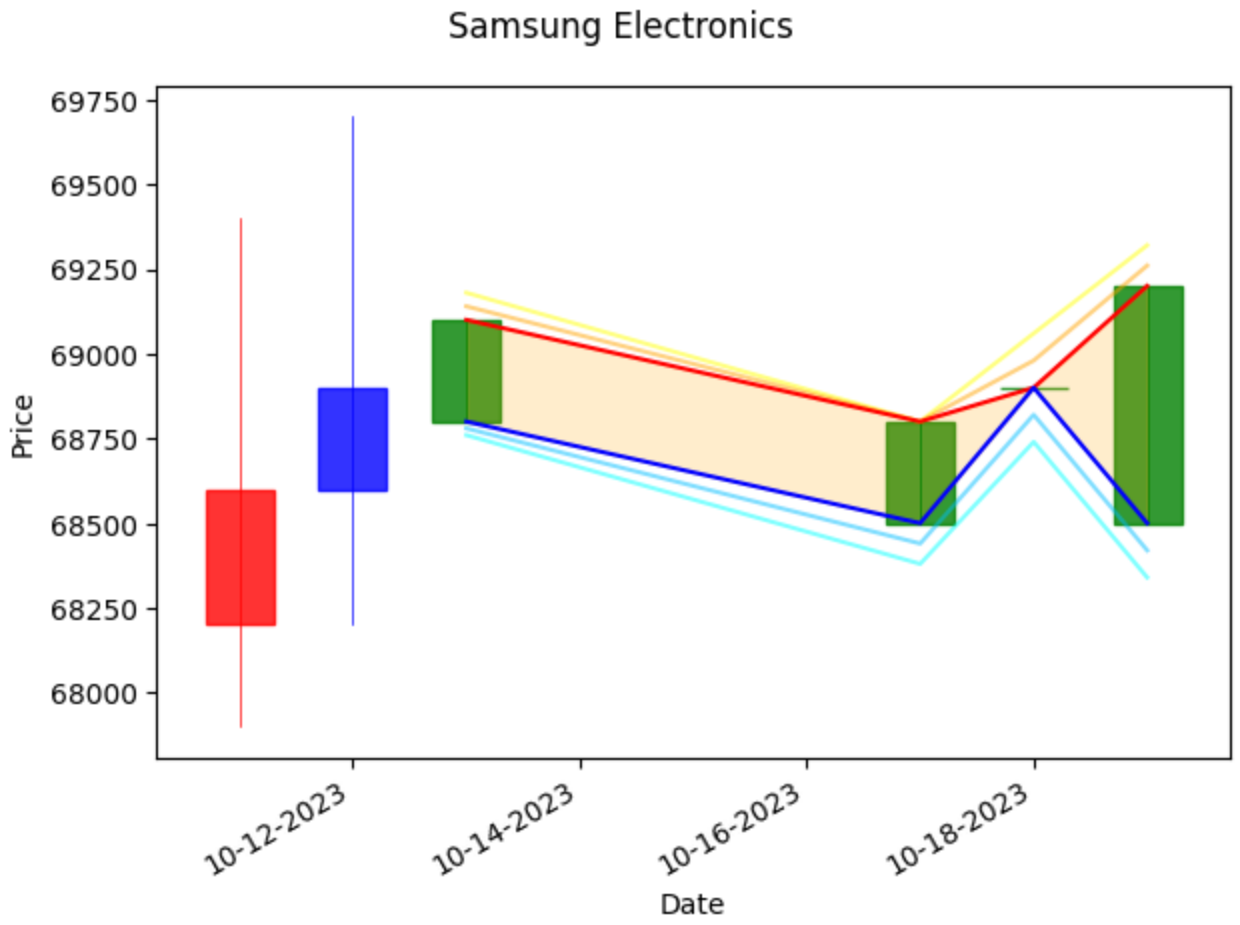
Front-end Progress : Django Installation


```
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 6.2.0-1013-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Thu Oct 12 11:45:35 UTC 2023

  System load:  0.0751953125      Processes:             98
  Usage of /:   36.3% of 7.57GB   Users logged in:       0
  Memory usage: 21%               IPv4 address for eth0: 10.10.137.72
  Swap usage:   0%

 * Ubuntu Pro delivers the most comprehensive open source security and
   compliance features.

   https://ubuntu.com/aws/pro

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status


Last login: Thu Oct 12 11:45:35 2023 from 203.252.33.2
ubuntu@ip-10-10-137-72:~$ sudo pip3 install django
pip3 install --upgrade django
Collecting django
  Using cached Django-4.2.6-py3-none-any.whl (8.0 MB)
Requirement already satisfied: asgiref<4,>=3.6.0 in /usr/local/lib/python3.10/dis
Requirement already satisfied: sqlparse>=0.3.1 in /usr/local/lib/python3.10/dist-
Requirement already satisfied: typing-extensions>=4 in /usr/local/lib/python3.10/
Installing collected packages: django
Successfully installed django-4.2.6
WARNING: Running pip as the 'root' user can result in broken permissions and conf
https://pip.pypa.io/warnings/venv
Defaulting to user installation because normal site-packages is not writeable
Requirement already satisfied: django in /usr/local/lib/python3.10/dist-packages
Requirement already satisfied: sqlparse>=0.3.1 in /usr/local/lib/python3.10/dist-
Requirement already satisfied: asgiref<4,>=3.6.0 in /usr/local/lib/python3.10/dis
Requirement already satisfied: typing-extensions>=4 in /usr/local/lib/python3.10/
ubuntu@ip-10-10-137-72:~$
```

Front-end Progress : Test Plot of Stock Price



Front-end Progress : Test Plot of Stock Price



Back-end Progress: Data class for train and test

Train Date : 2000~2022

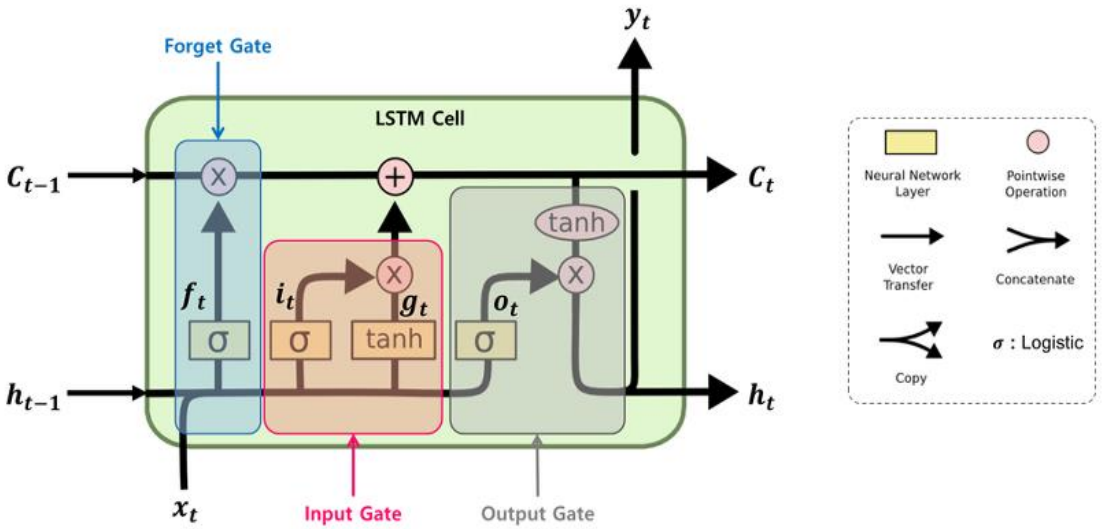
Test Data : 2023

```
class FinanceDataset(Dataset):
    def __init__(self, data_args, mode='train'):
        ...
        if self.stock_id == 'samsung':
            if mode == 'train':
                df = fdr.DataReader('005930','2000','2022')
            elif mode == 'test':
                df = fdr.DataReader('005930','2022','2023')
        ...
    df = df[['Open', 'High', 'Low', 'Volume', 'Close']]
    scaler = MinMaxScaler()
    df = scaler.fit_transform(df)
```

Back-end Progress : Implementation LSTM

LSTM : Long Short-Term Memory

Solves gradient vanishing problem of RNN when input data is too long



Back-end Progress : Implementation LSTM

```
class FinanceLSTM(nn.Module):
    def __init__(self, model_args):
        super(FinanceLSTM, self).__init__()

        self.output_length = model_args.output_length
        self.num_layers = model_args.num_layers
        self.input_size = model_args.input_size
        self.hidden_size = model_args.hidden_size
        self.fc_hidden_size = model_args.fc_hidden_size
        self.dropout = model_args.dropout

        self.lstm = nn.LSTM(input_size = self.input_size, hidden_size = self.hidden_size,
                            num_layers = self.num_layers, dropout = self.dropout, batch_first = True)
        self.fc1 = nn.Linear(self.hidden_size, self.fc_hidden_size)
        self.fc2 = nn.Linear(self.fc_hidden_size, self.output_length)
        self.relu = nn.ReLU()
```

Back-end Progress : Implementation LSTM

```
class FinanceLSTM(nn.Module):
...
    def forward(self, x):

        h_0 = torch.Tensor(torch.zeros(self.num_layers, x.size(0), self.hidden_size))
        c_0 = torch.Tensor(torch.zeros(self.num_layers, x.size(0), self.hidden_size))

        output, (hn, cn) = self.lstm(x, (h_0, c_0))
        hn = hn.view(-1, self.hidden_size)
        logits = self.relu(hn)
        logits = self.fc1(logits)
        logits = self.relu(logits)
        logits = self.fc2(logits)

        return logits
```

Plans for Next Progess Meeting

Donghun Jung

- Creating the initial UI/UX design

Chanyoung Lee, Yujin Seo

- Implementing GRU, CNN, Transformer